PERFORMANCE ANALYSIS TO SUPPORT B2C SYSTEM IN AIRLINE INDONESIA BASED ON SOA USING ENTERPRISE SERVICE BUS

Finda Anisa Putri, Abba Suganda Girsang
Computer Science Department, BINUS Graduate Program-Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia 11480
Email: finda.putri@binus.ac.id, agirsang@binus.edu

ABSTRACT

System airline as one of business to customer (B2C) is an enterprise application which needs service system architecture. Many systems that have been connected to this system has complexity of integration. SOA using enterprise service bus (ESB) is one way to manage the integration of information systems. ESB is a software infrastructure that can solve complexity problems with n-to-n integration, where n is the number of integrated applications. The business case has been implemented each function of services and ESB. The results obtained show that each services and ESB has its own strengths and weakness. As regards strengths, it should be noted that ESB copes very well with both kinds of files and small and large size but put the effort if the application of ESB is not working well it can be affected to another application system. N-to-n integration performs as good enough with small files and size but need to manage if there were any add or updated with the service. This research implemented SOA using ESB to proves to be a good choice for integration solution in a business environment with a large number of users and diversified communications.

Key words: Enterprise Service Bus, SOA, Airline, Performance, Integration.

1. INTRODUCTION

The digital era is now starting to lead to online transactions, one of the most popular web-based platforms in this era and very important is electronic commerce (e-commerce). E-commerce is about using computer system and internet to propose the business including selling, buying, exchanging products, servicing and information (Ueasangkomsate, 2015). In recent decades, e-commerce has developed rapidly as a series of dynamically growing technologies, where applications and business are radically transferred into digital. The airline industry is one of the industries that has long started selling tickets via online compared to
other industries (Arridha, Mohamed, & Elias, 2014). Starting to lead to online transactions, as well as airline ticket booking. The use of the Internet as a booking method offers companies and consumers benefits by reducing costs and providing the latest information to both parties (Mouakket, 2014). Purchasing online airline ticket has developed using an online system or can be called with e-ticketing. This makes it easy for customers to be able to buy airplane tickets, giving savings to customers and the airlines themselves. Getting a lot of customers is one of the fundamentals of success in airlines, so the company cooperates with several companies to get benefits from each other. Successful service companies recognize the relevance of cultivating loyal customers to increase sales and work to build long-term relationships with their customers through the membership in loyalty program (Vilkaitė-Vaitone & Papsiene, 2016). The Airlines also offer additional services, such as adding luggage, ordering food and selecting seats on the plane. These additional services can be called ancillary products. The ancillary product that can increase the profitability of an airline in the process of booking an online flight ticket can be said to be an ancillary revenue. Ancillary revenue has become a popular revenue stream (Uddin & Leon, 2017), because airlines look for ways to increase revenue by selling a variety of products or services when consumers travel on airplanes. Cooperates with several companies and providing ancillary products are parts of online reservation ticket in B2C that is connected to other applications via web services. Web services are able to function in program language standard using in development or its platform to communicate with or sending message with other applications or web services not based on standards and technologies of other platform (Platform Independence) (Kaewrattanapat & Nookhong, 2017). The application of web services for system of B2C has been developed on a n-to-n basis which is considered irrelevant because the number of services that continues to increase and become complex is difficult to control and manage. Different standards for web service applications are often also an obstacle if companies need integration and communication between web services B2C system must be interoperable to support enterprise business processes. Service oriented architecture helps us to develop concrete solution to achieve business processes flexibility. SOA is a logical way of designing system which provides services to either end-user or to other services distributed over a network via interfaces (Kavitha, Shanmugapriya, & Dhayalan, 2017). In contrast, an SOA can support the execution of an enterprise’s strategy in an environment with dynamic changes (Chang & Hwang, 2015). ESB (Enterprise Service Bus) has been created in order to enable the integration and interoperation of different applications, services or other resources within an organization (Górski & Pietrasik, 2017). Enterprise Service Bus (ESB), a form of centralized service by utilizing the SOA architecture that combines several web services in a container in the form of ESB to facilitate applications with different platforms to communicate with each other in accessing data (RajKumar & Vinod, 2015). In this article, every service connected in a B2C system is analyzed by the work test results using ESB and n-to-n. Some parameters measured in this case are services with different message types and transmission times for access to messages. The tools used to measure this are SOAP-UI with speed units in milliseconds.

1.1. Related Studies
In this paper, B2C system built a model architecture based on Service Oriented Architecture. Currently the application active using n-to-n web services that connected for many application. Modification the system of B2C with establish the service using Enterprise Service Bus which could make some comparison performance for the booking system and get the design for service implementation.
1.2. Service Oriented Architecture and Enterprise Service Bus

Agility, cooperation and efficiency are the most important factors in influencing the success of a company. These factors determine the capabilities of companies to respond to changing business needs. Traditional fixed and centralized infrastructure does not meet the company to build an application integration that continues to grow for the needs of the business world and can be a difficult, inefficient and expensive system (Sri & Priyadarsini, 2015). Service-oriented Architecture (SOA) is a kind of architecture model and a set of design methodology, which aims to maximize the reuse of neutral services in application to improve IT adaptability and efficiency (You, Xu, & Wang, 2016). The key conception of SOA is services, each application of SOA is regarded as service for calling and managing. The SOA defines the design principles: The functions are divided into less and reused model by using modularization. The clients and servers of loosing coupling do not require the close dependence. The encapsulation encloses function modularization (Wei, 2015). There are three roles in SOA, namely service brokers, service providers and service requester. The framework for SOA is depicted as a figure 1.

Service broker register and classifications of published service providers, also provide search services. The service provider will issue its own service and also respond to the request. Service applicants look for services that are needed by using a service broker and utilizing these services. Find Operations to help service applicants find special service assistance by service brokers. Operation of providers to help service providers in registering their own functions and interfaces. Bind operation to help service applicants in using services that are distinguished in the deed SOA service communication with messages formally designed via Extensible Markup Language (XML) to send or receive messages and without limitation by operating systems or connected programming languages. Under the guidance of SOA services, the construction of corporate informatization to develop the source of the company's production with other standards into services, the company's business processes are built by the choreography of services. Services are a type of SOA implementation. SOA components interact with each other. Some standard techniques are used in web services such as Universal Description, Discovery and Integration (UDDI), Web Services Description Language (WSDL), HyperText Transfer Protocol (HTTP), Simple Object Access Protocol (SOAP), and XML and so on. Web services are the best choice for developing SOA-based applications. The process of designing software systems based on SOA, the dynamic relationship of requirements for functions and services obtained, such as the relationship between the services and the bottom technology of the implemented services. Subsequently, basic services must meet the requirements of the singular-language definition and function in SOA. So each individual implementation service often depends on other application systems. Finally, the service process in an organization is used to implement all system functions. Service Oriented Architecture (SOA) is a strategy of enterprise business integration, which emphasis on reuse and loose coupling. As the core of SOA, the Enterprise Service Bus (ESB) is designed to integrate heterogeneous platform applications, provide interactive, collaborative, and
combined web-based distributed bus services for SOA. This article explores a cross-platform, cross-system application integration solution based on ESB to deal with the issues facing the enterprise informatization under the traditional architectures. The messages integrated through ESB need to be translated, subscribed, published and sent. The types of messages can contain decomposition of information, notification of information, registration of information and storage of information etc. The principle is: through the standard integration technology, SOA, Web Service and XML technologies are fused into a unified distributed architecture, which are easy to deploy and manageable (Ma, Yao, & Shan, 2017). The difference from the solution that uses ESB is publishes the service address by building the necessary modifications without changing the technical way in the data section associated with the software. This address allows to use uniform standard naming. Figure 2 shows, companies are rebuilding existing business systems. The system achieves data interoperability through the interface. Likewise, the dependence between services and transformation costs can be reduced.

![Figure 2. Difference before and after using ESB of enterprise system](image)

### 1.3. Study Background

As a case study for testing the performance service bus, the order execution process in B2C application system in Indonesia Airline which connected with some service as ticketing, ancillary, partnership and payment. This web application has some features besides booking ticket, the interoperability of the following IT system has description:

![Figure 3. BPMN Reservation B2C Application System](image)
• Ticketing System - provide services to determine availability of flight schedules based on route, date and number of passengers. Continue with generate booking code based on Passenger Number Record as an unique code to identify the ordered.
• Ancillary System - Additional features in carrying out airline ticket purchase transactions such as buying luggage, food and flexible seat selection
• Partnership System - Partnership that has worked with this airline to reserve tickets by making a profit in accordance with the agreement.
• Payment System - Payment of ticket and ancillary product reservations can be made with several payment methods available.

Figure 3 presents a Business Process Modelling Notation Diagram (BPMN) of the B2C system in one of Indonesia Airline system with integrated IT system.

The customer makes a ticket reservation in B2C application system. In other way, Customer from member of partnership get some benefits based on agreement both of the airline company and the partnership while booking the ticket. Indonesia airline offers the ancillary product to consumer to get an add value for extra service as extra baggage, seat selection and meals. Customer could make a payment with that transaction with some payment methods as ATM, e-banking or credit card. After the ticket was issued, customer get the ticket with an unique booking code as an identity reservation ticket. Each system that is interconnected uses different types of services such as partnerships that have cooperated to provide services with SOAP or REST types, ticket reservations, ancillary products and generate booking codes using SOAP. Whereas payment can use SOAP or REST as a service based on the available payment method. All differences from the types of services that cause integration between these systems use n-to-n based on services.

2. PROPOSED METHOD

All services are available on this B2C application system, along with the environment that has been run and tested on a n-to-n configuration and using the Enterprise Service Bus method. Figure 4 show architecture ESB in B2C application system.

![Figure 4. Architecture ESB in B2C Application System](http://www.iaeme.com/IJCIET/index.asp)
Every system connected with the B2C system application shown in Figure 4 requires communication between the payment system, the ancillary system and the partnership system. The communication process is supported by SOA using an enterprise service bus. Usually, every IT system connected to a B2C application is different in terms of the data format used. Data transferred between IT systems is usually in XML and JSON formats. Therefore the need to configure the flow of integration between systems that participate in data exchange. In the case of an online flight ticket reservation, the input message translated from NuSOAP to XML is done when the customer makes flight ticket reservations from starting to search for available flights and buying ancillary product transactions shown in figure 5. Then Figure 6 shows when the customer chooses the payment method then the payment system send data with wrap in SOAP. Then the payment gateway can translate input from NuSOAP to XML or REST to JSON depending on the availability of data integration on the payment service company. Likewise with the picture shown in Figure 7 of the partnership system that cooperates with this online ticket reservation company, they can provide data in various format. In this case, the partnership using REST to JSON for send and respond the data.
Build B2C application system using Enterprise Service Bus, SOA as a technology that is tested for service validation and service functions. So as to produce a ticket reservation system and additional products with the Enterprise Service Bus concept that can produce applications in accordance with business developments. In this journal, the application system from B2C will compare the speed of \( n \)-to-\( n \) integration performance using ESB as shown in the following figure 8.

![Figure 8. Performance Comparison ESB and \( n \)-to-\( n \) Integration](image)

3. RESULTS AND DISCUSSION

In this case it is used for performance testing: Ticketing, Ancillary, Payment and Partnership system. Each case is prepared for each system. In total there were 11 test cases prepared. Table 1 specifies a test case with service names implemented with the response time in milliseconds (ms).

<table>
<thead>
<tr>
<th>No.</th>
<th>Services</th>
<th>Descriptions</th>
<th>Systems</th>
<th>Response Time (ms)</th>
<th>ESB</th>
<th>( N )-to-( n )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Route Destination</td>
<td>List of all route destinations</td>
<td>Ticketing</td>
<td>2.899</td>
<td>2.919</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Searching Flight</td>
<td>Show flight schedule</td>
<td></td>
<td>3.204</td>
<td>4.097</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Generate Passenger Number Record</td>
<td>Create booking code as an unique reservation</td>
<td></td>
<td>500</td>
<td>594</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Retrieve Passenger Number Record</td>
<td>Show summary reservation flight booking</td>
<td></td>
<td>1.367</td>
<td>1.482</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>List of Baggage</td>
<td>List of available baggage</td>
<td>Ancillary</td>
<td>3.657</td>
<td>3.800</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>List of Seat Selection</td>
<td>List of available seat in a plane</td>
<td></td>
<td>807</td>
<td>931</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>List of Meals</td>
<td>List of available meals in a plane</td>
<td></td>
<td>1.167</td>
<td>1.668</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Assign Seat Selection</td>
<td>Booking seat selection</td>
<td></td>
<td>1.449</td>
<td>1.825</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Buy Add On</td>
<td>Booking baggage and meals selection</td>
<td></td>
<td>989</td>
<td>1.007</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Set Payment</td>
<td>Service of payment gateway</td>
<td>Payment</td>
<td>567</td>
<td>587</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Partnership</td>
<td>Service of partnership</td>
<td>Partnership</td>
<td>477</td>
<td>501</td>
<td></td>
</tr>
</tbody>
</table>
Getting the performance test steps for the integration solution, each test case listed in Table 1 is executed 5 times. The value of the performance measure is taken the lowest value for each test case. The number of messages there is not increased for these services. Using a monitor with a Windows system and a tool used to collect performance measurements is a SOAP UI configured in terms of the test cases listed in the table (Table 1). Displaying the performance measurement results of each case is presented in graphical form in figure 9 so that the difference between performance measurements using ESB and n-to-n integration is seen.

![Diagram Performance Analysis](image)

**Figure 9.** Diagram Response Time Performance Analysis

The results achieved by the Enterprise Service Bus in this performance testing should be highlighted. The utilization rate for each service that is run on n-to-n and ESB integration has different results. Measured speed in units of milliseconds (ms), in figure 9 the ESB results are shown in blue and n-to-n in yellow. The division means enough for the services that exist when run in the process performance. For Partnership, Set Payment, Buy Add On and Route Destination, although the difference in speed of performance does not show a significant difference, but when viewed from the fixed number, ESB outperforms performance in response, the time taken is 477 ms (ESB) and 501 ms (n-to-n) for partnerships, 567 ms (ESB) and 587 ms (n-to-n) for Set Payment, 989 ms (ESB) and 1,007 ms (n-to-n) for Buy Add On while Route Destination gets 2,899 ms (ESB) and 2,919 ms (n-to-n). The results of performance that are more visible in the difference in the Assign Seat Selection process reach 1,449 ms (ESB) and 1,829 ms (n-ton-n), for List of Meals the number is 1,167 ms (ESB) and 1,668 ms (n-to-n), 807 ms (ESB) and 937 ms (n-to-n) for List of Seat Selection, 3,657 ms (ESB) and 3,800 ms (n-to-n) for List of Baggage. Whereas for Retrieve and Generate Passenger Number Records, get the number 500 ms and 1,482 ms using ESB, 594 ms and 1,482 ms using n-to-n. The Searching Flight process on ticket reservations gets 3,204 ms (ESB) and 4,097 ms (n-to-n) results.

4. CONCLUSIONS

This paper presents test results from performance for integration solutions designed using ESB and n-to-n. Every integration solution is built in a case study of the B2C application system in one of Indonesia Airline. It should be noted, the integration platform using ESB on
companies can turn services into a software asset that can be managed quantitatively and increase service reuse. The test results for each service in ESB integration through the process proved to outperform in terms of performance it generates compared to n-to-n.

REFERENCES


http://www.iaeme.com/IJCIET/index.asp editor@iaeme.com